Adaptive All-Hexahedral Mesh Generation Based on A Hybrid Octree and Bubble Packing

Kangkang Hu, Jin Qian, and Yongjie Zhang*

Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Summary. In this paper, a novel adaptive all-hexahedral (all-hex) mesh generation algorithm is proposed based on a hybrid octree. This hybrid octree consists of polyhedral cells and each grid point is always shared by eight cells by using a cutting procedure. A dual mesh with all-hex elements is then extracted by analyzing each grid point. Next, sphere or ellipsoid bubble packing is applied to improve the mesh quality. The anisotropic property of ellipsoid bubbles is defined based on an input physical domain. This method generates all-hex meshes with good quality and allows a flexible control on mesh adaptation when compared to other algorithms.

Key words: hybrid octree, all-hexahedral mesh, adaptation control, bubble packing, isotropic and anisotropic mesh

1 Introduction

Several methods have been developed to meet the challenge of all-hex meshing. Among these methods, all-hex mesh generation using an octree data structure was developed rapidly due to its robustness and effectiveness. However, most developments lack a flexible control on mesh adaptation, and a great amount of propagation is required. In addition, anisotropic hex mesh generation is still an open problem. An effective method that can handle the anisotropy while preserving a good boundary is still lacking. In this paper, we develop an improved cutting procedure based on Maréchal's algorithm [1] to produce less number of elements and vertices. We first generate a primal hybrid octree with polyhedra by using the improved cutting procedure, and then generate a dual mesh from the hybrid octree which consists of all-hex elements. Next, we use bubble packing to warp the mesh vertices. If sphere bubbles are adopted, a good isotropic mesh can be generated; on the other hand, by choosing ellipsoid bubbles, anisotropic properties can be incorporated into the resulting mesh. The key contributions of our work include: (1) a novel hybrid octree structure is introduced, supporting a more flexible control of mesh adaptation than other methods; and (2) our algorithm can incorporate anisotropic property of an input physical domain into the mesh.

*Corresponding author. Tel: (412) 268-5332; Fax: (412) 268-3348; Email: *jessicaz@andrew.cmu.edu* (Y. Zhang).

2 Hybrid Octree and All-Hex Mesh Generation

Given a smooth boundary surface, a feature sensitive error function is firstly used to detect the main surface features [3]. This error function estimates the difference of the isosurface between two neighboring octree levels. Given an error tolerance ε, we refine cells with a larger error ($> \varepsilon$). The balancing and pairing rules are used to build a strongly balanced octree (quadtree in 2D) [1]. The balancing rule guarantees that the level difference between two neighboring octants is at most one. The pairing rule is embedded inside the balancing rule, that is, if an octant is subdivided to comply to the balancing rule, its siblings (the other seven octants belonging to the same parent) are subdivided along with it. In this paper, a new cutting algorithm is developed to eliminate all hanging nodes to build a hybrid octree.

2.1 Cutting Algorithm

In the strongly balanced quadtree, four quadrants belonging to the same parent form a quadtree block. A transition edge is formed when one of the neighbouring blocks is subdivided. A cutting procedure is used in this paper to eliminate hanging nodes and guarantee that each grid point in the resulting hybrid quadtree is always shared by four polygonal cells. Thanks to the pairing rule, we have only one unique configuration in 2D. It is a transition between two neighbouring blocks (white and blue), as shown in Fig. 1(a). When one of the blocks (the blue one) is subdivided, there is a transition edge (the green line) between the white block and the blue block. Red points hanging on the transition edge are two hanging nodes. Fig. 1(b) shows our cutting method. The subdivided block is cut to eliminate hanging nodes. Black points represent newly generated points through the cutting process. Our cutting method eliminates hanging nodes in the subdivided block with little propagation, and the transition edge is moved toward the subdivided block. Thanks to the pairing rule again, transition cases in different directions can be conducted independently. Fig. 1(c) shows a situation with two transition edges (green lines) along the green block. The cutting procedure is conducted independently in two directions, see Fig. 1(d).

In the strongly balanced octree, eight octants belonging to the same parent octant form an octree block. A transition case happens when neighbouring blocks are at different levels. There are five unique transition cases, including one transition case on face and four transition cases on edge. Fig. 2 shows all transition cases and their corresponding cutting processes. Fig. 2(a) shows the transition on face. Octants with the same color belong to the same block. Only half of each block close to the transition face is drawn. There is a transition face between two blocks since the blue block is subdivided, the red points represent hanging nodes on the transition face. Fig. 2(b-e) show four unique transition cases on edge. Each block edge is shared by four octree blocks. A transition case on the edge happens when four blocks sharing this edge are at different levels. Only one fourth of each block close to the transition edge

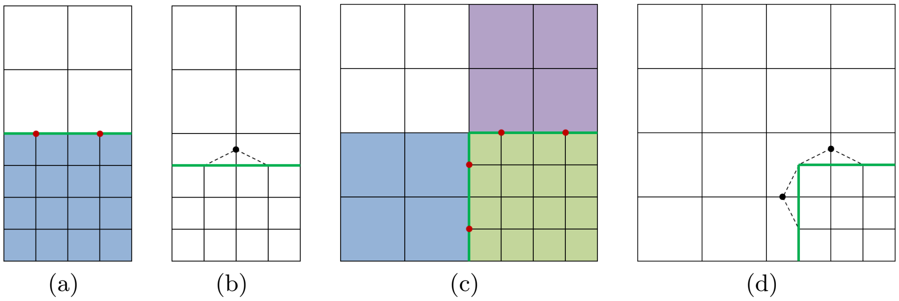

Fig. 1. (a-b) Two neighboring quadtree blocks and the cutting result; and (c-d) four neighbouring quadtree blocks and the cutting result.

is drawn, octants with the same color belong to the same block. To eliminate hanging nodes, our cutting method cuts in the subdivided blocks to guarantee all grid points in the obtained hybrid octree are always shared by eight polyhedral cells. Yellow planes represent the transition face, green lines represent the transition edge, red and blue dash lines represent cutting in different directions. Hanging nodes are firstly deleted, then the transition face and the transition edge are moved toward the subdivided block to reduce the propagation. Thanks to the pairing rule, the number of configurations is limited and the transition cases in each direction can be conducted independently.

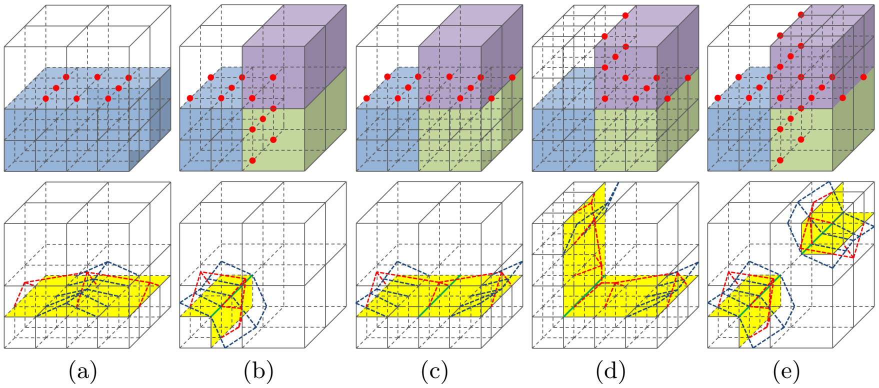

Fig. 2. First row: five unique transition cases; second row: the corresponding cutting processes. (a) Transition on face; and (b-e) transitions on edge.

2.2 Dual Mesh Generation

A dual all-hex mesh can be extracted from the obtained hybrid octree. For each primal grid point, one dual element can be formed by connecting dual vertices (here we use the cell center) of the primal cells sharing this grid point. Since all grid points in the hybrid octree are shared by eight polyhedral cells, the dual mesh of the hybrid octree is all-hex. After generating the dual mesh for the whole domain, elements outside or close to the boundary surface are firstly deleted to obtain a hex core mesh, the projection technique is then used to handle the boundary [3].

3 Mesh Quality Improvement

We choose the scaled Jacobian [3] as a metric and use bubble packing and optimization-based smoothing to improve the mesh quality. The sphere or ellipsoid bubbles can be defined using three principle directions and the corresponding radius in each direction. In this work, we compute the bubble size and orientation using the gradients of the input temperature field. Suppose the field is $f(x, y, z)$, and the gradient is $\nabla f(x, y, z)$. For a boundary vertex, we use the rotation matrix: $R = \left\{ u \ v \ n \right\}^T$, where n is the surface normal, u is the projection of vector ∇f onto the local tangent plane, and $v = n \times u$. The sizing matrix Λ is defined as:

$$\Lambda = \left\{ \begin{array}{ccc} h^{-2} & 0 & 0 \\ 0 & (\beta * h)^{-2} & 0 \\ 0 & 0 & h^{-2} \end{array} \right\}, \tag{1}$$

where β is a constant and h is the element size in the gradient direction. These specifications are only obtained on boundary cells, and the interior specifications can be computed using propagation. We choose a mass-spring-damper model to approximate the movement of the bubbles [2]. Each bubble is assigned a point mass, and we assume all bubbles have the same mass m and the same damping coefficient c. Without loss of generality, here we set them as 1.0. We use the following governing equation to control the motion of the bubbles: $mx_i'' + cx_i' = f_i$, where mx_i'' is the inertia term, cx_i' is the damping force which is proportional to the bubble velocity, and f_i is the summation of all inter-bubble forces acted upon this bubble.

4 Results and Conclusion

All results were computed on a PC equipped with a 2.93 GHz Intel X3470 CPU and 8GB of Memory. Fig. 3 shows isotropic and anisotropic effects on uniform and adaptive meshes of an igea model. A temperature field is given to define the bubble shapes. In red domains where the temperature is high, we want the mesh size to be smaller; vice versa. For anisotropic meshes, elements have a larger size along the contouring direction than along the gradient direction of the temperature domain. With bubble packing and optimization, we obtain good quality adaptive all-hex meshes with the minimum Jacobain of ≥ 0.15. In conclusion, a novel adaptive all-hex mesh generation algorithm is proposed based on a hybrid octree. Moreover, bubble packing and optimization-based smoothing are applied to improve the mesh quality. Our method has a flexible control of mesh adaptation and can incorporate anisotropic property of the input physical domain. In the future, we will apply the algorithm to more complicated models.

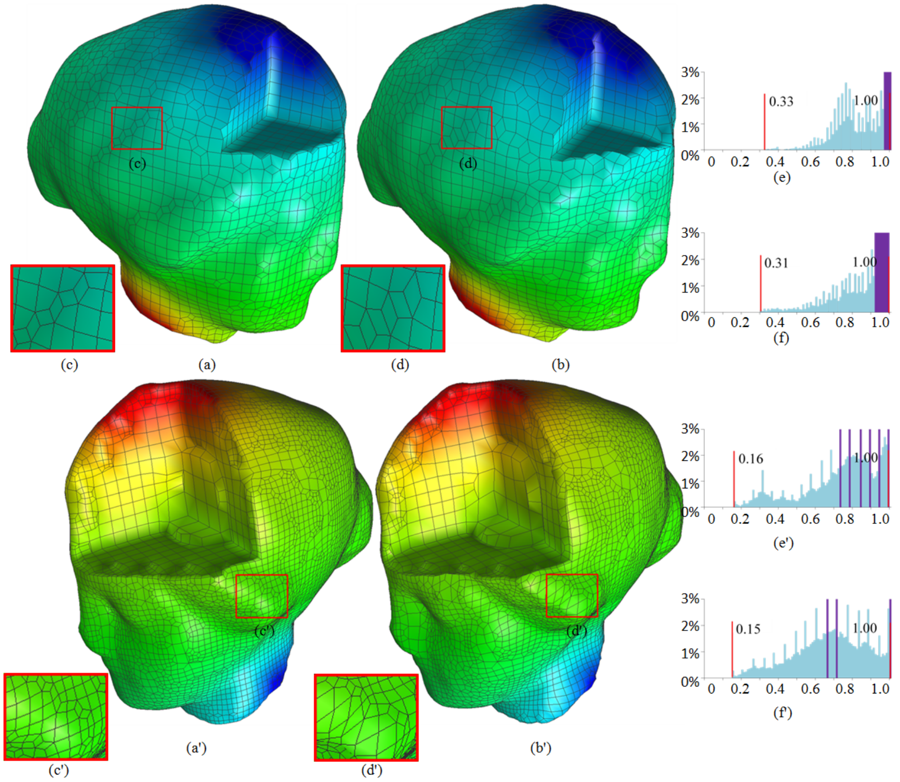

Fig. 3. Isotropic (a, c, e, a′, c′ and e′) and anisotropic (b, d, f, b′, d′ and f′) meshing results of an igea model. Top row: uniform meshes; bottom row: adaptive meshes; (a, b, a′ and b′) show mesh cross sections; (c, d, c′ and d′) show zoom-in details; (e, f, e′ and f′) show the scaled Jacobian histograms and purple bars are truncated ones due to a higher frequency ($\geq 3\%$).

References

[1] L. Maréchal. Advances in octree-based all-hexahedral mesh generation: handling sharp features. *Proceedings of the 18th International Meshing Roundtable*, pages 65–84, 2009.

[2] S. Yamakawa and K. Shimada. High quality anisotropic tetrahedral mesh generation via ellipsoidal bubble packing. *Proceedings of the 9th International Meshing Roundtable*, pages 263–273, 2000.

[3] Y. Zhang, X. Liang, and G. Xu. A robust 2-refinement algorithm in octree or rhombic dodecahedral tree based all-hexahedral mesh generation. *Computer Methods in Applied Mechanics and Engineering*, 256:88–100, 2013.